1-1-2004

Application of rough sets technique in resource selection problem

Qiang Meng
Iowa State University

Application of rough sets technique in resource selection problem

by

Qiang Meng

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Information Assurance

Program of Study Committee:
Dan Zhu, Major Professor
Prem Premkumar
Yuhong Yang

Iowa State University

Ames, Iowa

2004

Graduate College
Iowa State University

This is to certify that the master's thesis of

Qiang Meng

has met the thesis requirements of Iowa State University

Signatures have been redacted for privacy

TABLE OF CONTENTS

ABSTRACT iv

CHAPTER 1. THE RESOURCE SELECTION PROBLEM 1
Introduction 1
Hard/soft Constraints and Hard/soft attributes 2
Properties of Resource Selection Problem and Criteria for a Good Algorithm 3

CHAPTER 2. A SIMPLE-MINDED MODEL 9

CHAPTER 3. A TRADITIONAL ROUGH SETS MODEL 13
Basic Concepts 13
Set Approximations in Rough Sets 14
A Rough Sets Solution to the Resource Selection Problem 15

CHAPTER 4. A GENRALIZED ROUGH SETS MODEL 18
Limitations of traditional Rough Sets Theory 18
Generalized Rough Sets (GRS) Theory 20
A GRS Model for Resource Selection Problem 22

CHAPTER 5. A HIGH-ORDER ROUGH SETS MODEL 28
Two Types of Rules 28
Concepts and Notations in High-Order Rough Sets Theory 29
An HORS Resource Selection Model 32
A Simplified HORS model 34
Results and Analysis 36

CHAPTER 6. CONCLUSIONS 40

REFERENCES

ACKNOWLEDGEMENTS

ABSTRACT

In this paper a resource selection problem is studied. We analyze the properties of the resource selection problem and propose some criteria for a good resource selection model. A simple-minded model, a traditional rough set model, a generalized rough set (GRS) model, and a high-order rough set (HORS) model are introduced and the advantages and disadvantages of each model are compared. It is shown that the HORS model is superior to other models except that it is much more time consuming. Some methods to reduce the complexity of the HORS model are also proposed.

CHAPTER 1. THE RESOURCE SELECTION PROBLEM

Introduction

Resource selection is a crucial step in data intensive decision making processes. Generally a data intensive decision making process includes four steps: sourcing, resource selection, negotiation, and deciding. Resource selection involves screening items in the sources and generating a subset for human decision makers to use in the negotiation stage. In this paper we consider a software development project. The project is composed of m components. Suppose there are UDDI repositories that contain n software contractors. Let $U = \{u_1, u_2, \ldots, u_n\}$ denote the collection of all software contractors. Throughout this paper we will use $|S|$ to denote the number of elements in a set S. The UDDI repositories contain information about the contractors including location, yearly revenue, quality level, main type of components the contractor is capable of producing, and the types of programming languages the contractor has used. Meanwhile, each software component also has several attributes such as size, functional area, programming language, and budget. In this paper we use a simulated dataset consisting of 80 software contractors, see table 1.1 (in the end of this chapter). Furthermore we assume there are k human decision makers. For example, if an accounting information system is to be developed, then people involved in the contractor selection process might be the general manager, CEO, CFO, CTO, and the senior accountant. The purpose is to find suitable software contractors to send out call-for-quote so that it can acquire proposals that meet its software development needs. Due to the fact that there might be thousands of software contractors in the repositories, it would be too tedious and time consuming, if not impossible, to do this job manually. Therefore it is desirable to develop techniques to

generate a subset of U so that it is practical for human decision makers to screen every contractor in this subset and make final decision. In this paper we will present several models for the resource selection problem. First a simple-minded model is provided, and then we apply traditional rough sets model as well as VPRS and GRS models to this problem. Finally we propose a High-Order Rough Sets (HORS) model. Each model generates a subset U_s of U. Then the human decision makers screen every contractor in Us and make the final choice.

Hard/soft Constraints and Hard/soft attributes

A constraint is a resource selection rule specified by a human decision maker. Throughout this paper a constraint is also referred to as a rule. The concepts of hard and soft constraints are introduced in [Zhao, L.J. and Zhu, D. 2003]. Hard constraints are defined as "the ones that have to be met" while "soft constraints are relative loose constraints that can be somehow reduced or relaxed".

Given a constraint, it is not easy to define a quantity to measure how hard/soft it is. But we do have some guidelines for comparing the softness/hardness of two or more constraints.

1. Constraints on some certain attributes are likely to be harder than constraints on other attributes. For example, in the resource selection problem, a constraint on Language is likely to be harder than a constraint on Revenue. For instance, if the programming language java is required by the project due to certain hardware platform and operating system requirements then only those software contractors which support java should be selected. This constraint can not be relaxed. In contrast, the constraint on revenue is very likely to be a soft one. For example, suppose we require that a contractor should not be

selected if its yearly revenue is less than $250 million. But it is really not wise to reject a contractor with $240 million yearly revenue if all other attributes perfectly match our requirements, especially when no contractor can match all requirements.

2. If there are many parameters needed to be set subjectively by human decision makers then this constraint is likely to be a soft one. This is easy to understand. The reason is that the softness/hardness measures how likely a constraint can be changed. If a constraint contains many parameters that are determined subjectively, then these parameters are likely to be changed therefore the constraint itself is likely to be changed.

Similarly we can define hard and soft attributes. If the value of an attribute is likely to change significantly within a short period of time (1 year, for instance), then this attribute is called a soft attribute, otherwise it is called a hard attribute. For example, a company is unlikely to move to different states or countries frequently while its revenue of this year is always different to that of last year. Therefore the attribute *location* is hard while the attribute *revenue* is soft.

The existence of soft constraints and soft attributes implies that resource selection is a probabilistic instead of a deterministic problem. Accordingly, a good resource selection algorithm should take this characteristic into consideration.

Properties of Resource Selection Problem and Criteria for a Good Algorithm

The resource selection problem can be considered as a classification problem. Because selecting a subset from the available contractors is equivalent to classifying the contractors as

two categories – 1 (selected) or 0 (not selected). In most other classification problems, a model is applied to a set of data and an accuracy rate can be calculated based on the output of the model and the actual classification of each instance. Therefore one can evaluate a model based on its accuracy rate and other quantities such as false positive rate, false negative rate, etc. However, in the resource selection problem, this can not be done because there is no "actual" or "correct" classification. This makes it difficult to evaluate a model. The customers' choice can be used as the actual classification. However this method is flawed because the customers' choice is not always the best choice. In fact in most cases, customers have no enough time and energy to screen every product or travel agency and choose the best one. For this reason, a high accuracy rate does not imply the model's outcome is the best. Instead, it only means that the model's outcome is similar to customers' choice. One alternative for customers' choice is experts' opinion, i.e., experts' opinion is considered to be always correct. In our problem, suppose there are many projects, for each project a group of experts screen every contractor and select the best one. Then we can compute the percentage of projects for which the best contractor is included in the *Us* generated by a model. This percentage can be considered as the accuracy rate of the model.

Unfortunately, at present experts' opinion is not available so we have to find other criteria to decide the quality of a model. In this paper we propose the following criteria for evaluating the quality of resource selection models.

1. *Practicality*. The number of contractors in U_s must be reasonable. For example, it is practical for human decision makers to investigate ten contractors while it is considered

unreasonable if there are one hundred contractors in U_s. And clearly U_s should not be empty. If these conditions hold, we say the model is *practical* otherwise it is *impractical.*

2. *Sensitivity*. The data in the UDDI repositories may change with time due to the existence of soft attributes. For example, a contractor's revenue may increase and its quality of service may be improved. The data may also change due to the existence of error. Moreover, some soft constraints may be relaxed or strengthened. It is reasonable to require that a good model should not be too *sensitive* to changes in data and rules. In other words, a minor change in data or rules should not cause a significant change in the output of a model. Generally speaking, if the value of the attribute "decision" has a jump at some critical point (e.g. revenue=250, quality=B, etc.) then the model is sensitive.
3. *Reliability*. Different models require different rules. As discussed in section 1.2, a soft rule is likely to be changed and change in rules may lead to change in output of a model, especially if the model is sensitive. For a good model, a subjective change in the rules should not lead to a dramatic change in the output. Therefore the fewer soft rules a model requires, the more *reliable* the result is.
4. *Accuracy*. Every suitable contractor should be included in U_s. If this condition hold, we say U_s is *accurate*, otherwise it is *inaccurate*. When can a model be inaccurate? Or in other words, when will a model omit a contractor when this contractor is actually satisfies all requirements? Let's consider the following scenario: the annual revenue of contractor A is $255 million however due to statistical error it appears to be $245 in the database. In this case if a model only selects contractors whose revenue is above $250 million, then contractor A will not be selected even though it actually satisfies the requirement. This example shows that if there are soft attributes and soft rules, and the model is sensitive,

then the model is likely to be inaccurate. Or conversely, an insensitive and reliable model is accurate.

5. *Complexity*. Finally we require that a good model should not be too time consuming, i.e., it should not be too *complex*.

In this paper, we will propose four models for the resource selection problem, namely, the simple-minded model, the traditional rough sets model, the GRS model, and the HORS model. The advantages and disadvantages of each model will also be discussed, based on the criteria discussed above.

Table 1.1 Simulated data set

CID	Location	Revenue	Quality	Component	.net	Java	C++	Cobol
1	CA	740	C	Large	X		X	
2	CA	734	B	Small		X	X	
3	CA	1030	A	Large		X	X	
4	CA	817	C	Large	X	X	X	
5	CA	406	C	Medium	X	X	X	
6	CA	325	B	Small	X	X	X	
7	CA	550	C	Medium	X	X	X	
8	CA	545	B	Large		X	X	X
9	CA	247	A	Medium			X	
10	CA	672	C	Small		X	X	X
11	CA	160	A	Medium	X	X	X	X
12	CA	280	C	Medium		X	X	X
13	CA	527	C	Large	X	X	X	X
14	CA	992	B	Small		X	X	
15	CA	146	D	Large	X	X	X	
16	CA	293	C	Large		X	X	
17	CA	145	B	Large	X		X	
18	CA	146	A	Medium		X		
19	GA	816	C	Large	X	X	X	
20	GA	423	A	Medium		X	X	
21	GA	834	B	Medium	X	X	X	
22	GA	128	C	Large	X	X	X	
23	GA	752	B	Small	X	X	X	
24	GA	396	C	Small	X		X	
25	GA	967	B	Small		X	X	
26	GA	233	B	Large	X	X	X	
27	GA	502	A	Large			X	
28	GA	732	D	Medium		X	X	
29	IA	1035	A	Small		X	X	
30	IA	387	D	Small			X	X
31	IA	882	D	Large	X			X
32	IA	129	A	Large	X		X	
33	IA	428	B	Medium		X	X	
34	NY	878	B	Medium	X		X	
35	NY	605	B	Large		X	X	
36	NY	1039	C	Medium	X		X	
37	NY	894	C	Large	X		X	
38	NY	52	C	Large		X	X	X
39	NY	215	B	Medium			X	X
40	NY	666	C	Medium		X	X	

Table 1.1 Simulated data set (Continued)

CID	Location	Revenue	Quality	Component	.net	Java	C++	Cobol
41	NY	982	A	Medium		X	X	X
42	NY	628	B	Medium		X	X	
43	NY	851	B	Small		X	X	
44	NY	54	D	Large	X			
45	NY	185	C	Large		X	X	
46	NY	867	A	Medium	X		X	
47	NY	303	D	Medium	X	X	X	
48	NY	100	C	Small		X	X	
49	NY	180	C	Medium		X	X	
50	NY	461	B	Large	X	X		X
51	TX	571	C	Large	X	X	X	X
52	TX	62	C	Large		X	X	
53	TX	210	D	Small		X	X	
54	TX	991	C	Large	X		X	
55	TX	930	A	Medium	X	X	X	
56	TX	777	C	Medium	X	X	X	
57	TX	294	B	Small				
58	TX	266	D	Medium	X	X	X	
59	TX	467	A	Medium		X	X	
60	TX	178	B	Small		X	X	
61	TX	454	C	Medium		X	X	
62	TX	388	A	Small			X	X
63	TX	618	B	Large	X		X	X
64	TX	126	D	Small	X		X	X
65	TX	373	C	Medium	X			
66	TX	738	C	Small	X	X	X	X
67	WA	297	C	Large				X
68	WA	845	C	Large	X		X	
69	WA	868	B	Medium				
70	WA	873	C	Large			X	
71	WA	82	B	Small		X		
72	WA	204	C	Medium		X	X	
73	WA	710	A	Small	X		X	
74	Non US	408	A	Large	X			X
75	Non US	358	B	Medium			X	X
76	Non US	685	C	Large	X	X	X	
77	Non US	319	D	Large	X		X	
78	Non US	901	C	Large			X	
79	Non US	797	B	Medium				X
80	Non US	586	B	Medium	X	X	X	

CHAPTER 2 A SIMPLE-MINDED MODEL

In this section a straightforward solution to the resource selection problem is given. First we consider the single component case, i.e., the project consists of only one software component. Suppose each human decision maker has a rule for contractor selection. For example, a rule might be "if location = US, quality = A or B, and language includes Java then it should be selected." Let DM_i be the ith human decision maker and R_i be the rule given by DM_i. Each R_i will produce a subset of U. Let U_i be the subset produced by R_i. It is conceivable that different decision makers may have different rules, which will generate different subsets. For example, the CFO may be more interested in the contractors' revenue while the general manager may set a higher cutoff threshold for quality. In the simple-minded model, it is assumed that opinions of different people are equally treated. The set $\boldsymbol{U_s}$ can then be determined by:

$$(0201) \quad U_s = \bigcap_{i=1}^{k} U_i = \{u_i \in U : u_i \text{ is selected by all decision makers}\}$$

If the set (0201) returns an empty set then $\boldsymbol{U_s}$ is determined by:

$$(0202) \quad U_s = \{u_i \in U : u_i \text{ is selected by } (k-1) \text{ decision makers}\}$$

If the set $\boldsymbol{U_s}$ is still empty we can further loosen the constraints and find the contractors selected by *(k-2)* people. It is not too unrealistic to assume that there is at least one contractor that satisfies at least one rule. Therefore by repeating the above procedure we will finally get a nonempty set.

We will use the simulated dataset to test the simple-minded algorithm. Assume that there are four human decision makers and their rules are listed as follows:

R_1: If revenue>250, quality=A, B, or C, and language includes java, then select

R_2: If language includes .net, java, and cobol, then select

R_3: If revenue>550 and quality=A, B, or C, then select

R_4: If quality=A or B and language includes java, then select

The results are shown below:

$U_1 = \{u_2, u_3, u_4, u_5, u_6, u_7, u_8, u_{10}, u_{12}, u_{13}, u_{14}, u_{16}, u_{19}, u_{20}, u_{21}, u_{23}, u_{25}, u_{29}, u_{33}, u_{35}, u_{40}, u_{41}, u_{42}, u_{43}, u_{50}, u_{51}, u_{55}, u_{56}, u_{59}, u_{61}, u_{66}, u_{76}, u_{80}\}$;

$U_2 = \{u_{11}, u_{13}, u_{50}, u_{51}, u_{66}\}$

$U_3 = \{u_1, u_2, u_3, u_4, u_{10}, u_{14}, u_{19}, u_{21}, u_{23}, u_{25}, u_{29}, u_{34}, u_{35}, u_{36}, u_{37}, u_{40}, u_{41}, u_{42}, u_{43}, u_{46}, u_{51}, u_{54}, u_{55}, u_{56}, u_{63}, u_{66}, u_{68}, u_{69}, u_{70}, u_{73}, u_{76}, u_{78}, u_{79}, u_{80}\}$

$U_4 = \{u_2, u_3, u_6, u_8, u_{11}, u_{14}, u_{18}, u_{20}, u_{21}, u_{23}, u_{25}, u_{26}, u_{29}, u_{33}, u_{35}, u_{41}, u_{42}, u_{43}, u_{50}, u_{55}, u_{59}, u_{60}, u_{71}, u_{80}\}$

$U_1 \cap U_2 \cap U_3 \cap U_4 = \phi$

$\{u_i \in U : u_i \text{ is selected by 3 decision makers}\}$

$= \{u_2, u_3, u_{14}, u_{21}, u_{23}, u_{25}, u_{29}, u_{35}, u_{41}, u_{42}, u_{43}, u_{50}, u_{51}, u_{55}, u_{66}, u_{80}\}$

Therefore $Us = \{u_2, u_3, u_{14}, u_{21}, u_{23}, u_{25}, u_{29}, u_{35}, u_{41}, u_{42}, u_{43}, u_{50}, u_{51}, u_{55}, u_{66}, u_{80}\}$

Next we will check the practicality, accuracy, stability, reliability, and complexity of the simple-minded model.

1. Us contains 16 contractors. At first glance this is not a large number. However, considering the fact that there might be thousands of contractors in the UDDI repositories, the simple-minded algorithm may produce an Us that contains too many contractors. More important, in the simple-minded solution, a tighter restriction does not necessarily produce a smaller set. Therefore the human decision makers have no idea about how many contractors will be selected by the simple-minded algorithm. To illustrate this, let's take u_{13} as example. u_{13} has the following properties: location=CA, revenue=527, quality=C, component=large, language=.net,java,C,cobol. In the above example, u_{13} is in U_1 and U_2 but not in U_3 and U_4. Now suppose the third and fourth rules have been relaxed:

   R_3: If revenue>450 and quality=A, B, or C, then select

   R_4: If quality=A, B or C and language includes java, then select

   Then u13 will be in both U3 and U4 and consequently $Us = U_1 \cap U_2 \cap U_3 \cap U_4 = \{u_{13}\}$. This means while the original rules generate a subset of 16 contractors, a less strict set of rules generate only 1 contractor. This example shows that if the human decision makers can not control the number of contractors in *Us* by adjusting their rules.

2. In the simple-minded model, the attribute "decision" has jumps at many critical points therefore it is very sensitive. As shown in the above discussion, if we relax some soft constraints, the output of the simple-minded model will be completely different. This implies that this model is extremely sensitive to the change of constraints. To check its sensitivity to the change of data, we also take u_{13} as example. If u_{13} is a rapidly developing company, it is reasonable to assume that its yearly revenue increased by 10% and its quality of service was improved by one level. This means, the revenue of u_{13} will be $580 million and the quality level will be 2. Therefore it should be selected by all four human

decision makers thus becomes the only contractor in *Us*. This example shows that the results of the simple-minded model will change dramatically due to the change of data of one contractor.

3. The simple-minded model requires some soft rules because the human decision makers need to set a threshold value for each attribute. Some of these threshold values are likely to be relaxed. No other parameters are needed to be determined subjectively so we may describe the degree of reliability of the simple-minded model as "medium".
4. Since the simple-minded model is sensitive and needs some soft rules, we can conclude that it is inaccurate.
5. The simple-minded model can generate the result almost immediately. It can be considered as a "real-time reaction" model.

CHAPTER 3 A TRADITIONAL ROUGH SETS MODEL

Basic Concepts

Rough sets theory was introduced by Zdzislaw Pawlak in 1982 [Pawlak 1982]. It extends classical set theory. In rough sets theory, an *information system* consists of *universe*, *attributes*, *values of attributes* and a *function*. Symbolically, an information system is represented as: $S =< U, Q, V, f >$ where the universe U is the collection of objects we are interested in. Q is the collection of all attributes. Q can be divided into two disjoint nonempty subsets C and D where C is the collection of all condition attributes and D is the collection of all decision attributes. For convenience, we require that a decision attribute can only be equal to 1 or 0. Q, C, and D have the following properties: $Q \neq \phi$, $C \neq \phi$, $D \neq \phi$, $C \cap D = \phi$, and $C \cup D = Q$. V is the collection of the domain of each attribute $q \in Q$. The function $f : U \times Q \rightarrow V$ assigns a unique value $u_i \in U$ to each attribute $q_i \in Q$ of the object $u_i \in U$.

Any subset of the universe $X_i \subseteq U$ is called a *concept* or a *category*. Any family of concepts is referred to as *knowledge* about U. In particular, if $X_1, X_2, ..., X_n$ are concepts about U and the following conditions hold: for $1 \leq i, j \leq n$, $X_i \neq \phi$, $X_i \cap X_j = \phi$ if $i \neq j$, and $\bigcup_{i=1}^{n} X_i = U$, then $P = \{X_1, X_2, ...X_n\}$ is called a *classification*, and each $X_i \in P$ is called an *equivalence class* of U. If R is the *equivalence relation* that partitions U into P, then each X_i is called an *R-elementary set*. The collection of all *R-elementary sets* P is also denoted by U/R. If a set $A \in U$ is a finite union of R-elementary sets then A is called a *R-definable*, or *R-*

exact set otherwise R is *R-undefinable*, or *R-rough* set. The doublet *(U,R)* is called an *approximation space.*

Set Approximations in Rough Sets

Given an approximation space *(U,R)*, any set X in U can be approximated by a union of R-elementary sets in two ways: first, X can be approximated by the union of all R-elementary sets each of which is a subset of X; second, X can be approximated by the union of all R-elementary sets each of which has nonempty intersection with X. The first approximation, defined by $\underline{R}X = \bigcup \{X_i \in U/R : X_i \subseteq X\}$ is called the *R-lower approximation*, or the *R-positive region* of X. The second approximation, defined by $\overline{R}X = \bigcup \{X_i \in U/R : X_i \cap X \neq \phi\}$, is called the *R-upper approximation* of X. The set given by $BN_R(X) = \overline{R}X - \underline{R}X$ is called the *R-boundary* of X. In other words, the lower approximation of X is the collection of all objects that can be classified as definitely belonging to X by using the equivalent relation R and the upper approximation of X is the collection of all objects that can be classified as possibly in X using the equivalent relation R.

It has been shown in [Pawlak 1991] that the approximations have the following properties:

1) X is R-definable if and only if $\underline{R}X = \overline{R}X$

2) $\underline{R}X \subseteq X \subseteq \overline{R}X$

3) $\underline{R}\phi = \overline{R}\phi = \phi$; $\underline{R}U = \overline{R}U = U$

4) $\overline{R}(X \cup Y) = \overline{R}X \cup \overline{R}Y$

5) $\underline{R}(X \cap Y) = \underline{R}X \cap \underline{R}Y$

6) $X \subseteq Y$ implies $\underline{R}X \subseteq \underline{R}Y$ and $\overline{R}X \subseteq \overline{R}Y$

7) $(\underline{R}X \cup \underline{R}Y) \subseteq \underline{R}(X \cup Y)$

8) $\overline{R}(X \cap Y) \subseteq \overline{R}X \cap \overline{R}Y$

9) $\underline{R}(-X) = -\underline{R}X$ and $\overline{R}(-X) = -\overline{R}X$ where $-X = U\text{-}X$

10) $\underline{R}\underline{R}X = \overline{R}\underline{R}X = \underline{R}X$

11) $\overline{R}\overline{R}X = \underline{R}\overline{R}X = \overline{R}X$

The precision of approximations can be evaluated using the *accuracy measure* α_R given by $\alpha_R(X) = \frac{card(\underline{R}X)}{card(\overline{R}X)}$ where *card(A)* represents the cardinality of A, i.e., the number of elements in set A. Clearly $0 \le \alpha_R(X) \le 1$ and $\alpha_R(X) = 1$ implies $\underline{R}X = \overline{R}X$, which in turn implies that the set X is R-definable.

A Rough Sets Solution to the Resource Selection Problem

In our traditional rough sets model, the universe U consists of $n \times k$ objects, where n is the number of contractors and k is the number of human decision makers. We arrange the data in the following way: the first k objects are corresponding to the first contractor. The next k objects are corresponding to the second contractor, …, and so on. In this way the $n \times k$ objects can be divided into n groups. Each group is corresponding to one contractor. In each group, the first object is corresponding to the first decision maker, the second object is corresponding to the second decision maker, …, and so on. The condition attributes are the

information about the contractors in the UDDI repositories. For our sample data, C = {*Location, Revenue, Quality, Component, Language*}. There is only one decision attribute *decision*, the value of which is *1* if selected or *0* if not selected. We use all condition attributes to form an equivalence relation. If two objects represents the same contractor then they has exactly the same values for condition attributes hence they are in the same elementary set. It is possible that two different contractors have exactly the same properties but for convenience we regard them as in two different elementary sets. This will not affect the result. Therefore there are n elementary sets, denoted by $X_1, X_2, ..., X_n$. Each elementary set consists of k objects. The output U_s of the rough sets model is the lower approximation $\underline{R}X$ of the set X that is the collection of all contractors with *decision = 1*.

It should be noticed that if $\underline{R}X$ is not empty then the rough sets model is equivalent to the simple-minded model in that they generate the same U_s given the same dataset. This can be seen from the equivalence of the following statements:

The ith contractor is in the U_s generated by the rough sets model

$\Leftrightarrow$ X_i is a subset of X

$\Leftrightarrow$ *Decision = 1* for all objects in X_i

$\Leftrightarrow$ The ith contractor is selected by all human decision makers

$\Leftrightarrow$ The ith contractor is in the U_s generated by the simple-minded model

Therefore the only difference between the simple-minded model and the rough sets model is that whenever $\underline{R}X$ is empty the simple-minded model returns the set containing the

contractors selected by most decision makers while the rough sets model returns an empty set. This difference brings to the rough sets model some advantages as well as some disadvantages. The disadvantage of the rough sets model is that it is more likely to be impractical because it might return an empty set while the simple-minded model always produce a non-empty set. The advantage of the rough sets model is that it is easier for the human decision makers to control the number of contractors in *Us* by adjusting the constraints. For instance, if *Us* is empty or there are too few contractors in Us, the human decision makers might relax some constraints. This can be done because in the rough sets model, the tighter the constraints are, the fewer contractors there are in *Us*, and vice versa.

CHAPTER 4 A GENRALIZED ROUGH SETS MODEL

From the discussion in the previous two sections, it can be seen that the traditional rough sets model, as well as the simple-minded model, has many weaknesses: it might return a subset that contains too many or too few contractors, it might omit some good contractors, and it is very sensitive to changes in data and constraints. These weaknesses should be attributed to the intrinsic limitations of the traditional rough sets theory. [hu et. al. GRS] addressed these limitations and proposed a generalized rough sets (GRS) model. In this section, we modify the GRS model and apply it to the resource selection problem. Advantages and disadvantages of this model are also discussed.

Limitations of Traditional Rough Sets Theory

1. In traditional rough sets model, all objects are treated as equally important. In the resource selection problem, this implies that the opinions of all human decision makers are treated equally, as we have seen in the previous sections. However in real world, it is possible that opinions of some people are more influential in the decision making process. For instance, if the project is an Accounting Information System, the opinion of the chief Accountant might be considered especially important. A remedy to this problem is to assign a weight to each human decision maker, as we will see in the GRS model.
2. The decision attribute in the traditional rough sets model is represented crisply. This means, the decision is either 1 or 0. In real applications there are times when it is too expensive or risky to make a straightforward yes-no decision. This is the main reason why

the model is so sensitive to the change in data and constraints. In order to solve this problem, some uncertainty factors should be associated with the decision.

3. In the original rough sets model, the lower approximation is defined based on strict set inclusion. A consequence is that all elementary sets in the boundary region are treated equally. In our example, the elementary set u_3 and the elementary set u_{71} (remember that each contractor is an elementary set consisting of 4 objects) are both in the boundary region. If we compare these two contractors (see the table below) it can be easily seen that u_3 is far better than u_{71} in that u_3 is a larger company, has higher quality level, and has experiences in more programming languages. However the traditional rough sets model can not distinguish one from the other.

Table 4.1 Comparison of u_3 and u_{71}

CID	Location	Revenue	Quality	Component	Language
u_3	CA	$1030 m	A	Large	Java, c
u_{71}	WA	$82m	B	Small	Java

The simple-minded model can partly solve this problem. It can find the contractors selected by most decision makers.

4. The traditional rough sets model simply returns all contractors in the lower approximation hence it can not control the number of contractors in *Us*. Probably this is not a big issue in other classification problems, but it is a weakness for a resource selection model because an improper number of contractors in *Us* will make the model impractical. With this in mind, we may consider the resource selection problem as an ordering problem instead of a

classification problem. The idea is: if we can sort all contractors by some criteria then no matter how many contractors we need we can always find such a subset.

The traditional rough sets theory can do nothing but classify the universe into three categories, namely, the lower approximation, the boundary, and others. Thus it is not useful for the purpose of sorting. Fortunately, the GRS model, proposed by Xiaohua Hu et. al., provides us a tool realize our goal. Next we will introduce the GRS theory and construct an ordering model for the resource selection problem.

Generalized Rough Sets (GRS) Theory

The GRS theory extends the concept of information system in traditional rough sets theory and introduces an *uncertain information system (UIS)* in which each object is assigned an *uncertainty degree du* and an *importance degree di*. So an UIS can be represented as: $UIS =< U, Q, V, f, du, di >$ where *U, Q, V,* and *f* have exactly the same meaning as in the traditional rough sets model. The uncertainty degree *du* is a real number ranging from 0 to 1 with 1. It represents the possibility that the decision attribute is equal to 1. The importance degree *di* is a positive number that represents how important the object is.

The GRS theory adapts the concept of *relative classification error* that was introduced by [Ziarko]. The main idea is to put some elementary sets in the boundary region into positive region or negative region based on some classification factors. For each elementary set X_k, define *classification ratios* by:

$$(4.1) \qquad C_P(X_k) = \frac{\sum_{x_j \in X_k} du_j \times di_j}{\sum_{x_j \in X_k} di_j} \quad \text{and} \quad C_N(X_k) = \frac{\sum_{x_j \in X_k} du_j \times (1 - di_j)}{\sum_{x_j \in X_k} di_j}$$

where *xj* is the *j*th object in UIS.

$C_P(X_k)$ and $C_N(X_k)$ represent the certainty to classify X_k in the positive region and negative region, respectively. The relative classification error of classifying an object in X_k to the positive region and that of classifying the object to the negative region are given by $1 - C_P(X_k)$ and $1 - C_N(X_k)$, respectively.

Set approximation in the traditional rough sets theory can be generalized in the following way. An elementary set X_k belongs to the positive region if and only if $C_P(X_k)$ is greater than or equal to a given precision level P_β, or belongs to the negative region if and only if $C_N(X_k)$ is greater than or equal to a given precision level N_β, otherwise X_k belongs to the boundary region. Hence lower approximation $\underline{R}X$ and upper approximation $\overline{R}X$ can be defined by:

$$\underline{R}X = \bigcup \{X_i \in U/R : C_P(X_i) \geq P_\beta\}$$

$$\overline{R}X = \bigcup \{X_i \in U/R : C_N(X_i) \geq N_\beta\}$$

If the classification factors P_β and N_β increase, it means that the positive and negative region will shrink and the boundary region will expand. Therefore the size of the positive

region, negative region, and boundary region can be controlled by adjusting these two factors.

A GRS Model for Resource Selection Problem

As we mentioned earlier, for the resource selection problem, ordering models are superior to classification models because human decision makers can control the number of contractors in Us generated by any ordering model. In the GRS model, classification ratios are defined for each elementary set. Since each contractor is regarded as an elementary set in the resource selection problem, we can sort the contractors by C_p that represents the certainty to select a contractor. Another advantage of the GRS model is that we can assign a weight (i.e., the importance degree) to each human decision maker thus the importance of each decision maker can be taken into consideration. In the original GRS model proposed by [hu et. al], the uncertainty degree *du* is only defined for the decision attribute. In our problem *du* can not be defined directly. Instead, the human decision makers assign a score to all possible values of each attribute and *du* is defined as the product of these scores.

It should be noticed that the simple-minded model or the traditional rough sets model can be considered as a special case of GRS model. For instance, rule 1 in the simple minded model (If revenue>250, quality=A, B, or C, and language includes java, then select) is equivalent to assigning scores as in the following table. We assume that if not specified, the score is always equal to 1.

Table 4.2 Scores equivalent to rule 1

Attribute	Revenue		Quality		Language	
Value	>250	≤250	A,B, or C	D	Including Java	Otherwise
Score	1	0	1	0	1	0

Next we will apply the GRS model to our simulated dataset. We assume that the decision makers associate some uncertainty factors with their decisions. Take the first decision maker for example. He or she feels that it is too risky to ignore those contractors with revenue between 150 and 250 thus a score between 0 and 1 is assigned. The scores of each attribute assigned by human decision makers are shown in table 4.3 (in the end of this chapter). Moreover, we assume that the importance factors of the human decision makers are 1, 1, 0.7, and 0.5, respectively. Then for each contractor, C_P can be obtained by formula (4.1). Finally we sort the contractors by C_P and the top 15 contractors are shown in table 4.4.

It can be seen from the result that:

1. The result is always practical since the human decision makers can select an arbitrary number of contractors for further consideration.
2. The GRS model is less sensitive than the simple-minded model. Compared to the simple-minded model, the GRS model eliminates some jumps of the decision attribute and reduces the magnitude of other jumps. For instance, in the GRS model, *dui* changes gradually from 0 to 1 as revenue increases. And *dui* jumps from 0 to 0.5, then from 0.5 to 1 as quality of service improves, while in the simple-minded model the decision attribute jumps from 0 directly to 1.

Table 4.4 Top 15 contractors generated by GRS model

CID	du1	du2	du3	du4	Cp
51	1	1	1	0.8	0.958824
66	1	1	1	0.8	0.958824
13	1	1	0.923333	0.8	0.936275
2	1	0.5	1	1	0.897059
3	1	0.5	1	1	0.897059
14	1	0.5	1	1	0.897059
21	1	0.5	1	1	0.897059
23	1	0.5	1	1	0.897059
25	1	0.5	1	1	0.897059
29	1	0.5	1	1	0.897059
35	1	0.5	1	1	0.897059
41	1	0.5	1	1	0.897059
42	1	0.5	1	1	0.897059
43	1	0.5	1	1	0.897059
55	1	0.5	1	1	0.897059

3. A disadvantage of the GRS model is that there are too many parameters needed to be subjectively determined by human decision makers. This makes the rules in the GRS model softer than those in the simple-minded model. For this reason the GRS model is not reliable. This can be illustrated by the following example. Suppose the decision makers slightly change the scores they assigned for the GRS model, as shown in table 4.5 (in the end of this chapter). By "slightly change" we mean that for a fixed decision maker and a fixed attribute, the change of rule is not significant. For example, a change in score from 0.5 to 0.7, or a change in the threshold value from 550 to 600 is considered a slight change. The results are shown in table 4.6. It can be seen that the top 15 contractors selected by the GRS model change significantly. For instance, neither of the top 2 contractors in table 4.4 is included in the top 15 contractors in table 4.6.

Table 4.6 Top 15 contractors after change in parameters

CID	du1	du2	du3	du4	Cp
41	1	1	1	1	1
55	1	1	1	1	1
3	1	0.5	1	1	0.897059
29	1	0.5	1	1	0.897059
21	0.75	1	0.5	1	0.779412
23	0.75	1	0.5	1	0.779412
80	0.75	1	0.472333	1	0.771275
8	0.75	1	0.389481	1	0.746906
59	0.891321	0.5	0.469585	1	0.70909
2	0.75	0.5	0.5	1	0.676471
14	0.75	0.5	0.5	1	0.676471
25	0.75	0.5	0.5	1	0.676471
35	0.75	0.5	0.5	1	0.676471
42	0.75	0.5	0.5	1	0.676471
43	0.75	0.5	0.5	1	0.676471

4. Although the GRS model is not as sensitive as the simple-minded model, it requires many soft rules. Therefore the GRS model is not accurate.

5. Although more complex than the simple-minded model, the GRS model can also be considered as a "real-time reaction" model.

Table 4.3 Parameters in GRS model

Attribute	Decision maker 1		Decision maker 2		Decision maker 3		Decision maker 4	
	Value	Score	Value	Score	Value	Score	Value	Score
Revenue	(0,150]	0	Unspecified		(0,250]	0	Unspecified	
	(150,350]	Linear*			(250,550]	Linear*		
	(350,1050]	1			(550,1050]	1		
Quality	A,B, or C	1	Unspecified		A,B, or C	1	A or B	1
							C	0.8
	D	0.5			D	0.5	D	0
Language	Including Java	1	Including java and two of .net, c and cobol	1	Unspecified		Including Java	1
			Not including Java	0				
	otherwise	0.5	Otherwise	0.5			otherwise	0

* "Linear" means the value of score is linearly interpolated. For example, if score is linear for revenue between 150 and 250, then

$$score = \frac{revenue - 150}{250 - 150}$$

Table 4.5 Change in parameters

Attribute	Decision maker 1		Decision maker 2		Decision maker 3		Decision maker 4	
	Value	Score	Value	Score	Value	Score	Value	Score
Revenue	(0,150]	0	Unspecified		(0,350]	0	Unspecified	
	(150,350]	Linear*			(350,600]	Linear*		
	(350,1050]	1			(600,1050]	1		
Quality	A	1	Unspecified		A	1	A or B	1
	B	0.75			B or C	0.5		
	C	0.5					C	0.5
	D	0.25			D	0	D	0
Language	Including Java	1	Including Java, .net and cobol	1	Unspecified		Including Java	1
			Including none of Java, .net and cobol	0				
	otherwise	0.5	Otherwise	0.5			otherwise	0

CHAPTER 5. A HIGH-ORDER ROUGH SETS MODEL

Two Types of Rules

Generally in a data mining problem, there are two types of rules [Yao]. The first type of rules, called type 1 rules, focus on a single object. In the simple-minded model, traditional rough sets model, and GRS model, all rules involved are type 1 rules because they are all in the following form, "If the condition attributes take value a, then the decision attribute takes value b". The second type, called type 2 rules, focus on a pair of objects. A standard type 2 rule is in the form, "If two objects have the same value on attribute A, then they have the same value on attribute B". Type 2 rules are also referred to as *high order rules* because they represent a higher level of knowledge. A standard type 2 rule is sometimes called a *dependency rule* since it reflects the dependency relationship between two attributes. There are some variants of standard type 2 rule. By using a similarity relation we can obtain a *weak dependency rule* which is in the form "If two objects have similar values on attribute A then they have similar values on attribute B". An *ordering rule* can be obtained by introducing a preference relation. The form of an ordering rule is "If an object is ranked ahead of another object according to one set of attributes then the pair are ranked in the same way with respect to another set of attributes". (see [Yao]). In this paper we will investigate ordering rules in detail and present an ordering model for the resource selection problem. It should be noticed that although the GRS model is also an ordering model, it does not require any ordering rules.

Concepts and Notations in High-Order Rough Sets Theory

A *formula*, denoted by ϕ, is an ordering relationship between two objects based on an attribute A. ϕ can be represented as:

$x \succ_A y$, or in short, $\succ_A$

where x and y are two objects and the symbol $\succ_A$ means "ahead of, based on attribute A". The meaning set of a formula, denoted by $m(\phi)$ is the collection of all pairs of objects *(x,y)* such that x and y are in the universe and *(x,y)* satisfies the ordering relationship ϕ. The cardinality, i.e., the number of elements in $m(\phi)$, is denoted by $|m(\phi)|$. If neither *(x,y)* nor *(y,x)* is in $m(\phi)$, we say that x and y are *indiscernible* by the relationship ϕ, denoted by $x \sim_A y$.

Suppose ϕ and ψ are formulas, we adopt the following notations:

$(x, y) \in m(\neg\phi)$ if and only if $(x, y) \notin m(\phi)$

$(x, y) \in m(\phi \wedge \psi)$ if and only if $(x, y) \in m(\phi)$ and $(x, y) \in m(\psi)$

$(x, y) \in m(\phi \vee \psi)$ if and only if $(x, y) \in m(\phi)$ or $(x, y) \in m(\psi)$

For the meaning sets of ϕ and ψ, the following properties hold:

(1) $m(\neg\phi) = U \times U - m(\phi)$

(2) $m(\phi \wedge \psi) = m(\phi) \cap m(\psi)$

(3) $m(\phi \vee \psi) = m(\phi) \cup m(\psi)$

If $\phi = x \succ_A y$ and $\psi = x \succ_B y$, an ordering relationship between attribute A and B can be represented as: $\phi \Rightarrow \psi$, which can be interpreted as the logical implication. However in real world this interpretation may be too restrictive to be useful. Therefore probabilistic interpretations may be more practical. [Yao 23] suggests many such interpretations. Two measures, *accuracy* and *coverage*, are used in our paper. These two measures are defined as follows:

$$accuracy(\phi \Rightarrow \psi) = \frac{|m(\phi \wedge \psi)|}{m(\phi)} \text{ and}$$

$$\text{cov}erage(\phi \Rightarrow \psi) = \frac{|m(\phi \wedge \psi)|}{m(\psi)}$$

The accuracy measures the correctness of the rule and the coverage reflects the applicability of the rule. Generally speaking these two measures are not independent since both are related to the quantity $|m(\phi \wedge \psi)|$. In many cases there is a trade-off between them. This means, a rule with high accuracy may have a low coverage while a rule with high coverage may have a low accuracy.

To illustrate these concepts let's look at an example. Consider the following information about five computers:

Table 5.1 An illustrative example

	CPU(GHz)	Price	Warranty	Overall
1	2.5	$600	2 years	1
2	2.0	$800	2 years	3
3	3.0	$800	2 years	3
4	3.0	$700	2 years	2
5	3.0	$600	1 year	3

Define the following ordering relationships:

$\phi_1 =\succ_{CPU}: 3.0 \succ_{CPU} 2.5 \succ_{CPU} 2.0$

$\phi_2 =\succ_{Price}: 600 \succ_{Price} 700 \succ_{Price} 800$

$\phi_3 =\succ_{Warranty}: 2 \text{ years} \succ_{Warranty} 1 \text{ year}$

$\psi =\succ_{Overall}: 1 \succ_{Overall} 2 \succ_{Overall} 3$

Then the meaning sets of each relationship are:

$m(\phi_1) = \{(1,2),(3,1),(3,2),(4,1),(4,2),(5,1),(5,2)\}$

$m(\phi_2) = \{(1,2),(1,3),(1,4),(4,2),(4,3),(5,2),(5,3),(5,4)\}$

$m(\phi_3) = \{(1,5),(2,5),(3,5),(4,5)\}$

$m(\psi) = \{(1,2),(1,3),(1,4),(1,5),(4,2),(4,3),(4,5)\}$

Our purpose is to find the accuracy and coverage of the ordering rules $\phi_1 \Rightarrow \psi$, $\phi_2 \Rightarrow \psi$, and $\phi_3 \Rightarrow \psi$. First notice that

$m(\phi_1) \cap m(\psi) = \{(1,2),(4,2)\}$

$m(\phi_2) \cap m(\psi) = \{(1,2),(1,3),(1,4),(4,2),(4,3)\}$ and

$m(\phi_3) \cap m(\psi) = \{(1,5),(4,5)\}$

Therefore

$$accuracy(\phi_1 \Rightarrow \psi) = \frac{|m(\phi_1 \wedge \psi)|}{m(\phi_1)} = \frac{2}{7}$$

$$\text{cov}erage(\phi_1 \Rightarrow \psi) = \frac{|m(\phi_1 \wedge \psi)|}{m(\psi)} = \frac{2}{7}$$

$$accuracy(\phi_2 \Rightarrow \psi) = \frac{|m(\phi_2 \wedge \psi)|}{m(\phi_2)} = \frac{5}{8}$$

$$coverage(\phi_2 \Rightarrow \psi) = \frac{|m(\phi_2 \wedge \psi)|}{m(\psi)} = \frac{5}{7}$$

$$accuracy(\phi_3 \Rightarrow \psi) = \frac{|m(\phi_3 \wedge \psi)|}{m(\phi_3)} = \frac{2}{4}$$

$$coverage(\phi_3 \Rightarrow \psi) = \frac{|m(\phi_3 \wedge \psi)|}{m(\psi)} = \frac{2}{7}$$

It can be seen from the above results that the price of a computer has a closest relationship with its overall ranking because the ordering rule $\phi_2 \Rightarrow \psi$ has the highest accuracy and coverage.

An HORS Resource Selection Model

As we discussed in previous chapters, the GRS model is accurate, practical, and not very sensitive to changes in data. However, one major disadvantage of the GRS model is that there are too many quantitative parameters needed to be determined by human decision makers, and the determination of these parameters is quite subjective. In other words, the rules in the GRS model are very soft. To improve the reliability of the model we need to find some hard rules to be used in the model. Motivated by this idea and the fact that some high order rules can be considered very hard, we propose an HORS (High Order Rough Set) model in this section.

To illustrate why we assert that some high order rules are hard and how we can use these rules to construct a resource selection model, let's look at the following rules:

Rule 1 – If Quality = D then score=0, if Quality = B or C then score=0.5, if Quality = A then score=1

Rule 2 – If Quality = A,B or C then the contractor should be selected

Rule 3 – If the Quality of contractor u_1 is better than that of u_2 then u_1 should rank ahead of u_2.

Clearly rule 1 is very soft since both threshold values and the corresponding scores are subjectively determined by human makers, rule 2 is harder because only a threshold value needs to be set, and rule 3 is the hardest since no parameters need to be subjectively determined. Actually if other attributes of u_1 and u_2 are the same then rule 3 is always true. For real world data, rule 3 may not be true for some contractor pair (u_1, u_2) due to the influence of other attributes on the overall ranking. But we can safely claim that if the overall ranking of contractors is reasonable then the accuracy and coverage of rule 3 should be large (close to 1).

For the resource selection problem in this paper, each contractor has several attributes. For each attribute we can define a rule similar to rule 3 in the above example. We notice that in the models described in previous chapters, the attributes Location and Component is not important. To be consistent we only consider rules corresponding to attributes Revenue, Quality, and Language in the HORS model, i.e., the ordering relationships are:

$\phi_1 \Rightarrow \succ_{Revenue}$: Revenue of $u_1 \succ_{Revenue}$ Revenue of u_2 if Revenue of u_1 > Revenue of u_2

$\phi_2 \Rightarrow \succ_{Quality}$: A $\succ_{Quality}$ B $\succ_{Quality}$ C $\succ_{Quality}$ D

$\phi_3 =\succ_{Language}$: Language of $u_1 \succ_{Language}$ Language of u_2 if Language of u_1 includes Language of u_2

$\psi =\succ_{Overall}$: $1 \succ_{Overall} 2 \succ_{Overall} 3$, etc.

Define the objective function by

$$Obj = \sum_{i=1}^{3} Accuracy(\phi_i \Rightarrow \psi) + \sum_{i=1}^{3} Coverage(\phi_i \Rightarrow \psi)$$

Where accuracy and coverage are defined in section 5.2. For each ranking, or permutation of contractors we can compute the value of *Obj*. Our goal is to find the ranking such that *Obj* is maximized.

A Simplified HORS model

In the HORS model, we do an exclusive search, i.e., we compute the objective function for every possible permutation of contractors in order to find the best one. If the number of contractors is n, then the number of permutations is $n!$, For example, if n=80, there will be $80! = 7.16 \times 10^{118}$ possible permutations. Moreover, the number of permutations increases extremely rapidly as n increases. For instance, if n increases from 80 to 81, the number of permutations will increase by 81 times. These facts show that it is time consuming, if not impossible, to apply the HORS model to solve the resource selection problem in real world. For this reason we propose a simplified version of HORS model, which can be summarized as a three-step selection and ranking procedure, as described in the following paragraph.

Step1 Contractor selection based on conservative simple rules.

In this step the human decision makers set a conservative cut-off value for each attribute and a set of contractors U' is generated where $U'=\{u_i$:the value of attribute of u_i is better than the cut-off value for each attribute$\}$. By "conservative" we mean that this rule should not be very strict so that every good contractor should be included in U'.

Step2 Contractor selection based on high-order rules

Suppose there are n' contractors in U' from which we want to choose *ns* contractors. For each contractor u_i in U', let

The overall ranking = 1 if u_i is selected

= 0 if u_i is not selected

Altogether there are $C(n',n_s)=\frac{n'!}{n_s!(n'-n_s)!}$ possible combinations. For each combination we can compute the objective function *obj*, which is defined in the previous section, and find the combination that maximizes *obj*.

It should be noticed that $C(n',n_s)$ can still be a very large number, in this case we introduce a sub-step method. Divide step 2 into several sub-steps. In the first sub-step, we select *m* contractors where $m_1 \le n_s$, in the second sub-step we select *m2* contractors from the rest $(n_s - m_1)$ contractors where $m_2 \le n_s - m_1$, and so on. For large n', this will drastically reduce the time needed for this step. Suppose we divide this step into 2 sub-steps and let N_1, N_2 be the number of possible combinations before and after we do so. Moreover suppose we select *ns/2* contractors in each sub-step then we have

$$\frac{N_1}{N_2} = \frac{C(n', n_s)}{C(n', n_s/2) + C(n' - n_s/2, n_s/2)} \approx \frac{n'^{n_s/2}}{2 \times (n_s/2)!} > \frac{1}{2}\left(\frac{n'}{n_s/2}\right)^{n_s/2}$$

Consider that n' is usually much larger than $ns/2$, we conclude that this ratio is very large. In other words, we can dramatically reduce the number of combinations by dividing this step into 2 substeps.

Step3 Contractor ordering based on high-order rules

In this step we order the ns contractors selected in step2. This step is also based on the same high order rules ϕ_1, ϕ_2, ϕ_3, and ψ. For each permutation we compute the value of the objective function and find the permutation that maximizes this function. This step is not necessary if the human decision makers are planning to screen every contractor in Us.

Results and Analysis

Define the following conservative rules:

ϕ_0: If Quality>200, and Quality=A,B,or C, and Language include Java, then select

By applying this rule we obtained a subset U' of U:

U'={u_2, u_3, u_4, u_5, u_6, u_7, u_8, u_{10}, u_{12}, u_{13}, u_{14}, u_{16}, u_{19}, u_{20}, u_{21}, u_{23}, u_{25}, u_{26}, u_{29}, u_{33}, u_{35}, u_{40}, u_{41}, u_{42}, u_{43}, u_{50}, u_{51}, u_{55}, u_{56}, u_{59}, u_{61}, u_{66}, u_{72}, u_{76}, u_{80}}

The rule ϕ_0 is conservative in the sense that it is less strict compared with most rules in the simple-minded model and the GRS model. Actually all contractors selected by the simple-minded model and all top 15 contractors generated by the GRS model are in U'. There are 35 contractors in U' thus n'=35.

Next we apply the high-order rules ϕ_1, ϕ_2, ϕ_3, and ψ to U'. Here we assume that the human decision makers want to select 10 contractors for them to screen, i.e., ns=10. The second and the third step of the simplified HORS model are implemented using c++. The optimal selection of contractors is $Us = \{u_3, u_8, u_{14}, u_{21}, u_{23}, u_{29}, u_{41}, u_{55}, u_{66}, u_{80}\}$.

More details of the second and the third step are provided in the following part. Notice that there are a huge number of combinations and permutations of contractors, for this reason we only give the details of the optimal selection:

$$accuracy(\phi_1 \Rightarrow \psi) = \frac{|m(\phi_1 \wedge \psi)|}{|m(\phi_1)|} = \frac{213}{595} = 0.358$$

$$coverage(\phi_1 \Rightarrow \psi) = \frac{|m(\phi_1 \wedge \psi)|}{|m(\psi)|} = \frac{213}{250} = 0.852$$

$$accuracy(\phi_2 \Rightarrow \psi) = \frac{|m(\phi_2 \wedge \psi)|}{|m(\phi_2)|} = \frac{162}{384} = 0.422$$

$$coverage(\phi_2 \Rightarrow \psi) = \frac{|m(\phi_2 \wedge \psi)|}{|m(\psi)|} = \frac{162}{250} = 0.648$$

$$accuracy(\phi_3 \Rightarrow \psi) = \frac{|m(\phi_3 \wedge \psi)|}{|m(\phi_3)|} = \frac{95}{336} = 0.283$$

$$coverage(\phi_3 \Rightarrow \psi) = \frac{|m(\phi_3 \wedge \psi)|}{|m(\psi)|} = \frac{95}{250} = 0.380$$

Therefore

$$obj = 0.358 + 0.852 + 0.422 + 0.648 + 0.283 + 0.380 = 2.943$$

We assume that the human decision makers will screen every contractor in Us therefore the third step is not needed.

For comparison purpose we also tried the sub-step method. We first choose 5 contractors from the 35 contractors in *U'* then select another 5 contractors from the rest 30 contractors. The contractors selected in the first sub-step are: $\{u_3, u_{21}, u_{29}, u_{41}, u_{55}\}$ and the contractors selected in the second sub-step are: $\{u_8, u_{14}, u_{23}, u_{66}, u_{80}\}$ therefore we get exactly the same *Us*. This shows that the sub-step method will not significantly reduce the quality of the result.

Next we evaluate the HORS model based on the criteria proposed in section 1.3.

1. The HORS model is always practical because the number of contractors in the subset *Us* is determined by the human decision makers.
2. The HORS model is insensitive to change in data. The value of objective function changes gradually as the data changes. Therefore a slight change in data can only cause a slight change in the output of the HORS model. We are not interested in the sensitivity of the HORS model to change in rules since no soft rule is required in this model, as discussed below.
3. The HORS model is very reliable because the human decision makers only need to decide which attributes are important. No quantitative parameter is needed to be set.
4. As we have seen the HORS model is insensitive and reliable hence it is accurate.
5. Generally for the simplified HORS model, step 2 needs much more time than the other 2 steps. In our example, in step 2 there are $C(35,10) = 1.84 \times 10^9$ combinations while there are only $10! = 3.63 \times 10^6$ permutations in step 3. For this reason we only consider the complexity of second step. It takes more than 10 hours to finish the contractor selection

procedure (step 2) on a P4 2.0G PC. For the sub-step method, the number of combinations is:

$$C(35,5) + C(30,5) = 3.25 \times 10^5 + 1.43 \times 10^5 = 4.68 \times 10^5$$

and it only takes less than 1 minute to get the result. So we conclude that the HORS model is complicated but its complexity can be dramatically reduced if the sub-step method is used.

CHAPTER 6. CONCLUSIONS

1. In this paper we solved a resource selection problem utilizing the traditional rough sets, generalized rough set, and high-order rough sets techniques.
2. The resource selection problem has some special properties and it is hard to evaluate a resource selection model based on its accuracy rate. We proposed several criteria for a good model. In summary, a good model should be practical, insensitive, reliable, accurate, and not too complicated.
3. We introduced a simple-minded model, a traditional rough sets model (which is essentially equivalent to the simple-minded model), a GRS model, and a HORS model. Generally speaking the HORS model is superior to other models. The advantages and disadvantages are summarized in table 6.1.

Table 6.1 Comparison of resource selection models

	Advantages	Disadvantages
Simple-minded model and traditional rough set model	- Simple - Real-time reaction - Needs less soft rules than GRS model	- Can be impractical - Very sensitive to change in data and change in rules - Needs some soft rules - Can be inaccurate
GRS model	- Real-time reaction - Always practical - Less sensitive than simple-minded model	- Needs too many soft rules - Can be inaccurate
HORS model	- Always practical - Less sensitive than all other models - More accurate than all other models - Doesn't need soft rules	- Too complicated and time consuming but the complexity can be reduced to an acceptable level by using the simplified HORS model and sub-step method

REFERENCES

Bonikowski, Z., Bryniarski, E., Wybraniec-Skardowska, U., Extensions and intensions in the rough set theory, *Information Sciences*, **107,** 149-167, 1998.

Cohen, W.W., Schapire, R.E., and Singer, Y., Learning to order things, *Journal of Artificial Intelligence Research*, **10**, 243-270, 1999.

Fayyad,U., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R., Advances in knowledge discovery and data mining, AAAI Press / MIT Press, 1996.

Greco, S., Matarazzo, B., and Slowinski, R., Rough approximation of a preference relation by dominance relations, *European Journal of Operational Research*, **117**, 63-83, 1999.

Greco, S., Matarazzo, B., and Slowinski, R., Rough set theory for multicriteria decision analysis, *European Journal of Operational Research*, **129**, 1-47, 2001.

Hu, X.H., Cercone, N., Han, J. and Ziarko, W., GRS: A generalized rough sets model, *Data Mining, Rough Sets and Granular Computing*, Physica-Verlag, 447-460, 2002.

Inuiguchi, M. and Tanino, T., Two directions toward generalization of rough sets, *Rough Set Theory and Granular Computing*, Springer, 47-57, 2003.

Inuiguchi, M. and Tanino, T., On rough sets under generalized equivalence relations, *Bulletin of International Rough Set Society*, **5** (1/2), 167-171, 2001.

Intan, R., Yao, Y.Y., and Mukaidono, M., Generalization of rough sets using weak fuzzy similarity relations, *Rough Set Theory and Granular Computing*, Springer, 37-46, 2003.

Katzberg, J.D. and Ziarko, W., Variable precision rough sets with asymmetric bounds, *Proceedings of International Workshop on Rough Sets and Knowledge Discovery*, 163-190, 1993.

Lin, T.Y. and Cercone, N., Applications of rough sets theory and data mining, Kluwer Academic Publishers, 1997.

Pawlak, Z., Rough sets, *International Journal Computation & Information Science*, **11**, 341-356, 1982.

Pawlak, Z., Rough sets: theoretical aspects of reasoning about data, Kluwer Academic Publishers, 1991.

Pawlak, Z., Slowinski, R., Rough sets approach to multi-attribute decision analysis, *European Journal of operational Research*, **72**, 443-459, 1994.

Polkowski, L., Rough sets: mathematical foundations, Physica-Verlag, 2002.

Sai, Y., Yao, Y.Y. and Zhong, N., Data analysis and mining in ordered information tables, *Proceedings of 2001 IEEE Conference on Data Mining,* 497-504, 2001.

Slowinski, R. and Vanderpooten, D., A generalized definition of rough approximations based on similarity, *IEEE Transactions on Data and Knowledge Engineering,* **12**(2), 331-336.

Tsumoto, S., Automated discovery of plausible rules based on rough sets and rough inclusion, *Proceedings of 2001 IEEE Conference on Data Mining*, 497-504, 2001.

Yao, Y.Y., Mining high order decision rules, *Rough Set Theory and Granular Computing*, Springer, 125-135, 2003.

Yao, Y.Y. and Sai, Y., Mining ordering rules using rough set theory, *Bulletin of International Rough Set Society,* **5**, 99-106, 2001.

Yao, Y.Y. and Sai, Y., On mining ordering rules, *New Frontiers in Artificial Intelligence, Lecture Notes in Computer Science*, **2253**, Springer, 316-321, 2001.

Yao, Y.Y. and Zhong, N., An analysis of quantitative measures associated with rules, *Proceedings of PAKDD '99*, 479-488, 1999.

Zhao, L.J. and Zhu, D., Workflow Resource Selection from UDDI Repositories with Mobile Agents, *Proceedings of Web2003*, 2003.

Ziarko, W., Variable precision rough set model, *Journal of computer and system sciences,* vol. 46, No. 1, 39-59, 1993.

Ziarko, W., Analysis of uncertain information in the framework of variable precision rough sets, *Foundations of computing and decision sciences*, vol.18, No. 3-4, 381-396, 1993.

Ziarko, W., Rough sets, fuzzy sets and knowledge discovery, Springer-Verlag, 1994.